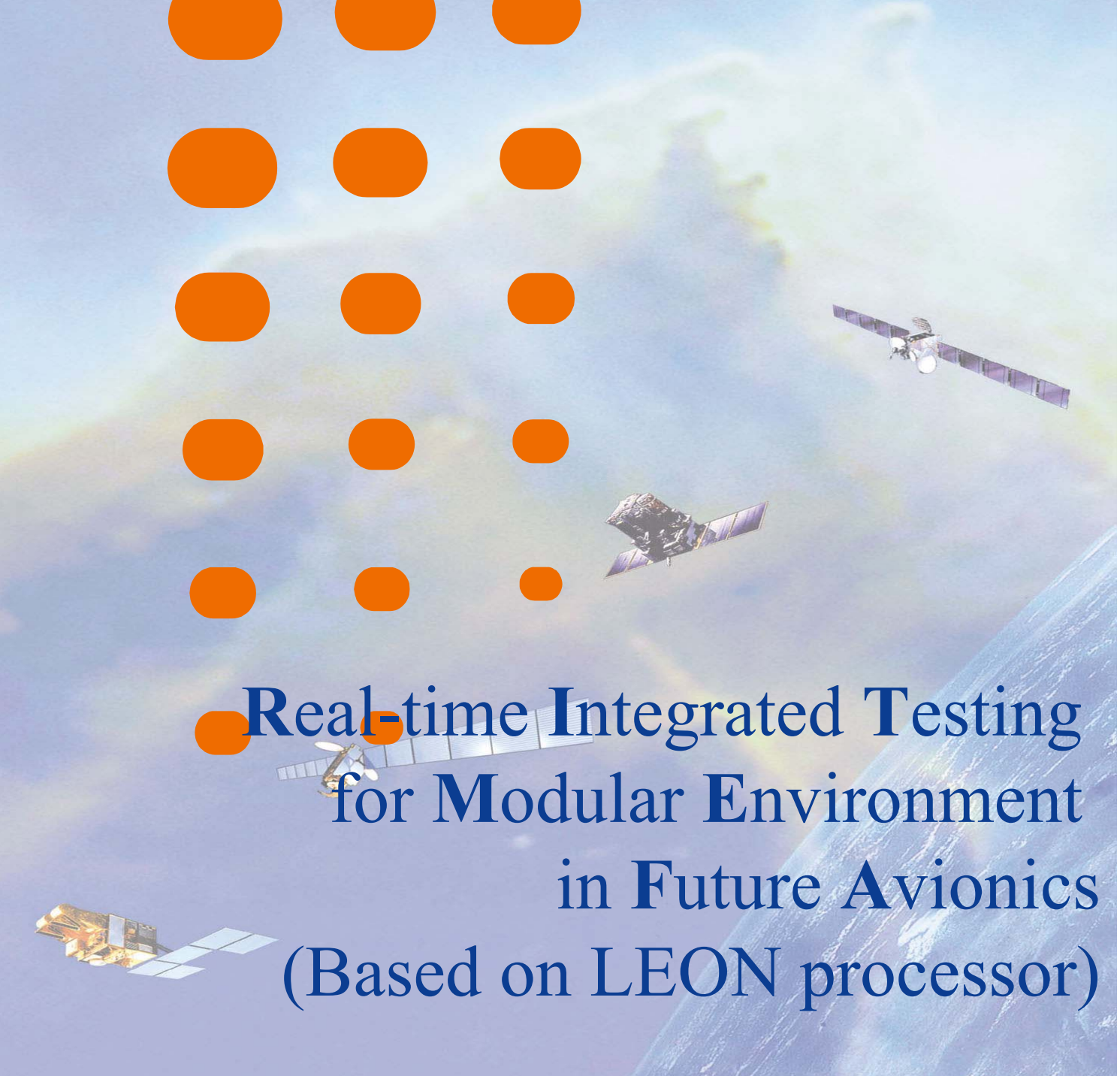


# Real-time Integrated Testing for Modular Environment in Future Avionics (Based on LEON processor)

astrium


Yves Dufrenne

Software Avionics Facilities expert
IA55

E-Mail:
yves.dufrenne@astrium-space.com


Nov 2002

# Objectives



## Codesign Hardware / Software based on LEON processor

- How to cope with cache based processors ?

- Change induced in current validation process.

- The LEON initiative in Astrium SAS.

# Contents



- **PAST : Flight software validation process in Astrium-SAS**
  - Flight software process & benches architecture
  - Feedbacks on these architecture & process
- **PRESENT : Solutions for the next avionic architectures**
  - What is LEON processor
  - Impacts in development process
- **FUTURE : Research & Development perspectives**
  - Architecture of a Real-Time bench with LEON inside
  - Roadmap

# The PAST

Intro

Past

Present

Future

- **PAST : Flight software validation process in Astrium-SAS**
  - Flight software process & benches architecture
  - Feedbacks on these architecture & process

- PRESENT : Solutions for the next avionic architectures
  - What is LEON processor
  - Impacts in development process

- FUTURE : Research & Development perspectives
  - Architecture of a Real-Time bench with LEON inside
  - Roadmap

# Flight Software Development Process

astrium

Intro

Past

Present

Future

## 5 Steps in development process

1) DHS FS debug & coverage in OL
2) AOCS FS test & performance in CL
3) Complete FS integration into a Real Target for timing & robustness tests
4) Target coupling with real equipments
5) System Validation and operators training for complete satellite

| | |
|---|---|
| **OL** | Open Loop |
| **CL** | Closed Loop |
| **FS** | Flight software |
| **DHS** | Data Handling System |
| **AOCS** | Attitude & Orbit Control System |

# Flight Software Development Process

Intro

Past

Present

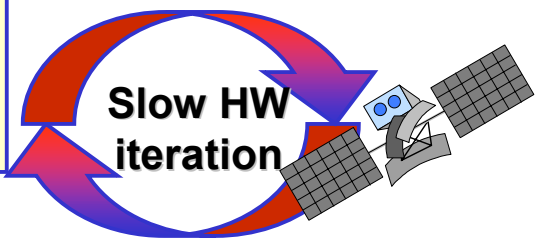
Future

**SVF** — **S**oftware **V**erification **F**acility
- Emulator on workstation
- Bottom half (DHS)
- None

**FVB** — **F**unctional **V**erification **B**ench
- Native + stubs on station
- Upper half (AOCS)
- Simple (AOCS Env.)

RITME USE

**SVB** — **S**oftware **V**alidation **B**ench
- Target bread-board
- Integrated (DHS+AOCS)
- Simple (Sometimes)




**AVB**
- Flat Sat. then platform
- Complete FS
- Nearly total (AOCS+Pow)

**A**ssembly **V**alidation. **B**ench

**OSB**
- Emulator
- Complete FS
- Total (AOCS+Pow+Therm)

**O**peration **S**imulation **B**ench

Onboard Calc.
Flight software
S/C Simulator

# Key lessons learned

Intro

Past

Present

Future

**SVF** — **S**oftware **V**erification **F**acility

Not totally realistic timings

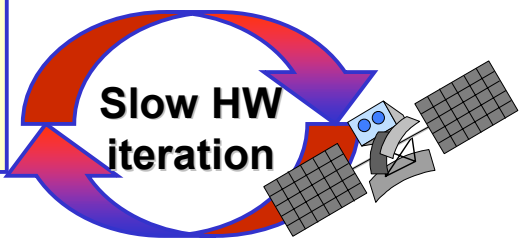
**FVB** — **F**unctional **V**erification **B**ench

Fast feedback between AOCS SW dvp & test is efficient

RITME USE

**SVB** — **S**oftware **V**alidation **B**ench

HW delivery too late, time consuming update




**AVB**

Need higher data rate transfer than telemetry

**A**ssembly **V**alidation. **B**ench

Use of a dedicated HW tool (MIA)

**OSB**

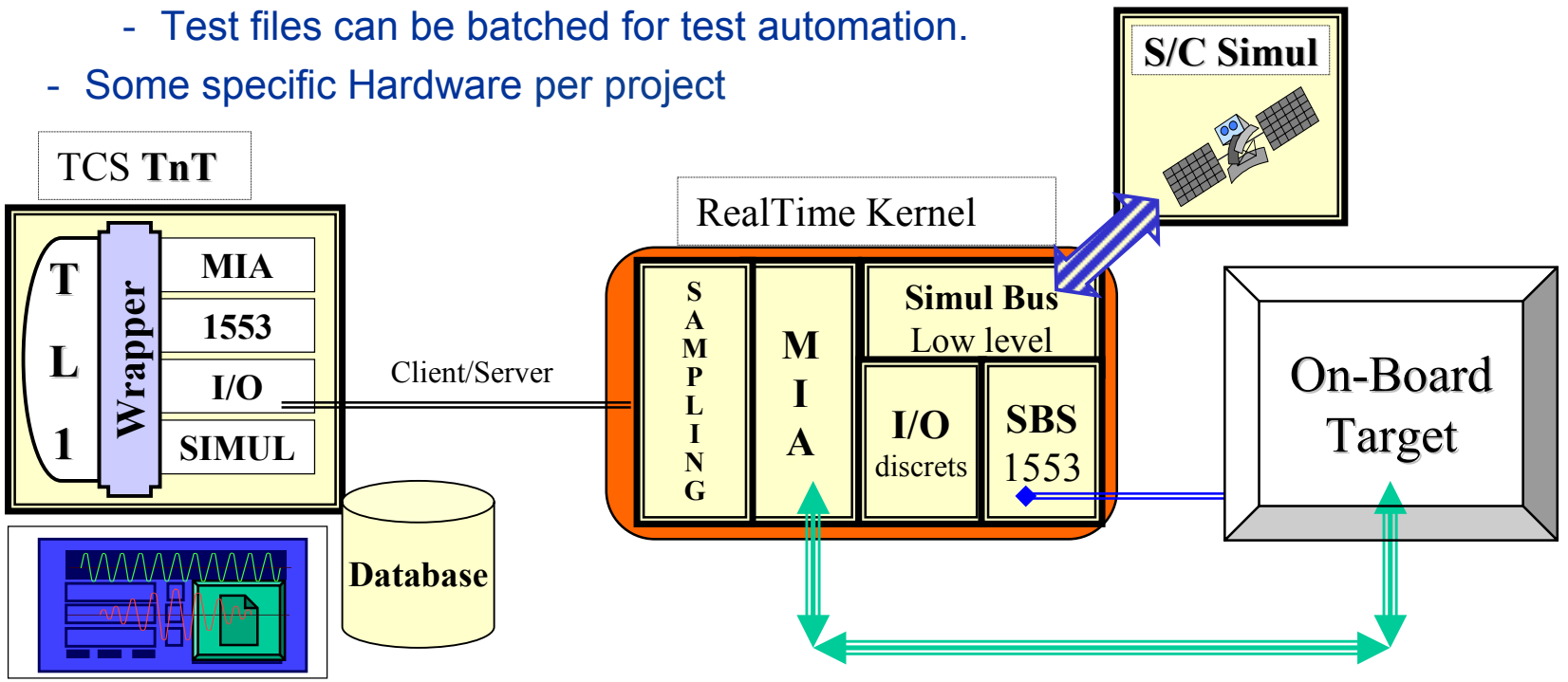
Better to reuse former benches' components than merge all needs

**O**peration **S**imulation **B**ench

# RITME Architecture

- **Real-time Integrated Testing for Modular Environment**
  - One or more Test Control Station (SUN under Solaris, but many functionalities already work under DEC OSF and PC Linux )
  - Set of software : TNT = Test Natural Toolkit
    - Library of drivers (1553, OBDH, 1355, MIA…)
    - Test language called TL1 with drivers symbolic plugging
    - MMI for monitoring and sampling symbols
    - Test files can be batched for test automation.
  - Some specific Hardware per project

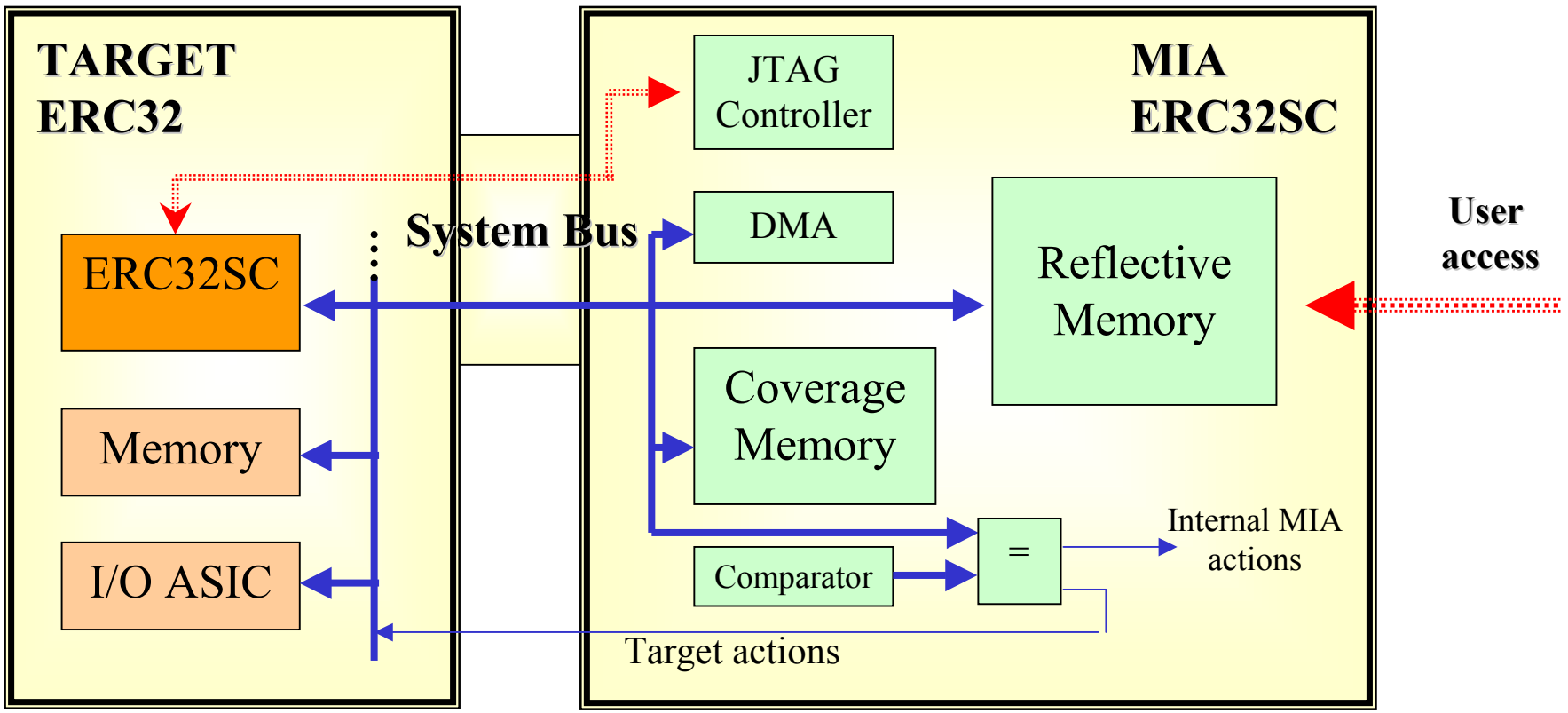
**MIA**
Micro
Interface
Analyser



# The MIA Answer

- MIA spy calculator system bus
- Non intrusive spy processor tools
- Use on both SVB & AVB

**MIA**
Micro
Interface
Analyser



# The PRESENT

Intro

Past

**Present**

Future

- PAST : Flight software validation process in Astrium-SAS
  - Flight software process & benches architecture
  - Feedback on these architecture & process

- **PRESENT : solutions for the next avionics architecture**
  - What is LEON processor
  - Impact in development process

- FUTURE : Research & Development perspectives
  - Architecture of a Real-Time bench with LEON inside
  - Roadmap

# LEON Processor



- **What is LEON ?**
  - On board processor developed by **ESA**,
  - **Open source VHDL** model provided by Gaisler Research
  - 32 bits compatible SPARC V8, **cached based** !!!
  - Integrate a **Debug Support Unit**
  - Set of tools (compiler, emulator,…)

# The LEON's impact



## Impacts

- Open source VHDL model processor
  => **Quick & representative target design on COTS board**
  Possible to add I/O on the same FPGA board
  VHDL Model allow to see inside processor

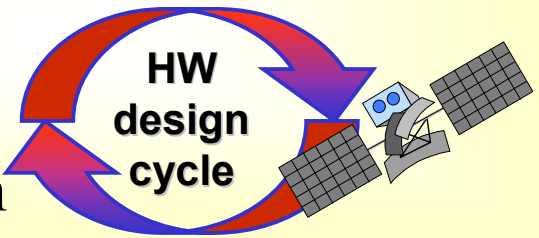
- External HW debug type MIA cannot see inside cache based processors
  => **Integrate debug system inside target**
  OK with DSU **intrusive** debug and trace.

- **But** our process need higher monitoring than Telemetry
  => Need to add **LIA** for **unintrusive reflective memory**

**LIA**
LEON
Interface
Analyser

# New Development Process

SVB ng
- Incremental flight software
- Simple S/C Simulation (AOCS Environment)
- FPGA with VHDL LEON model
- HW / SW Co-verification




Assembly Validation. Bench

AVB
- Flat Sat. then platform
- Complete FS
- Nearly total (AOCS+Pow)

Operation Simulation Bench

OSB
- **FPGA with LEON**
- Complete FS
- Total (AOCS+Pow+Therm)

Reuse as FS loader / debugger

Reuse as S/C operation simulator

# The FUTURE

Intro

Past

Present

Future

- PAST : Flight software validation process in Astrium-SAS
  - Flight software process & benches architecture
  - Feedbacks on these architecture & process

- PRESENT : Solutions for the next avionics architectures
  - What is LEON processor
  - Impact in development process

- **FUTURE : Research & Development perspectives**
  - Architecture of a Real-Time bench with LEON inside
  - Roadmap

# LEON SVB Architecture

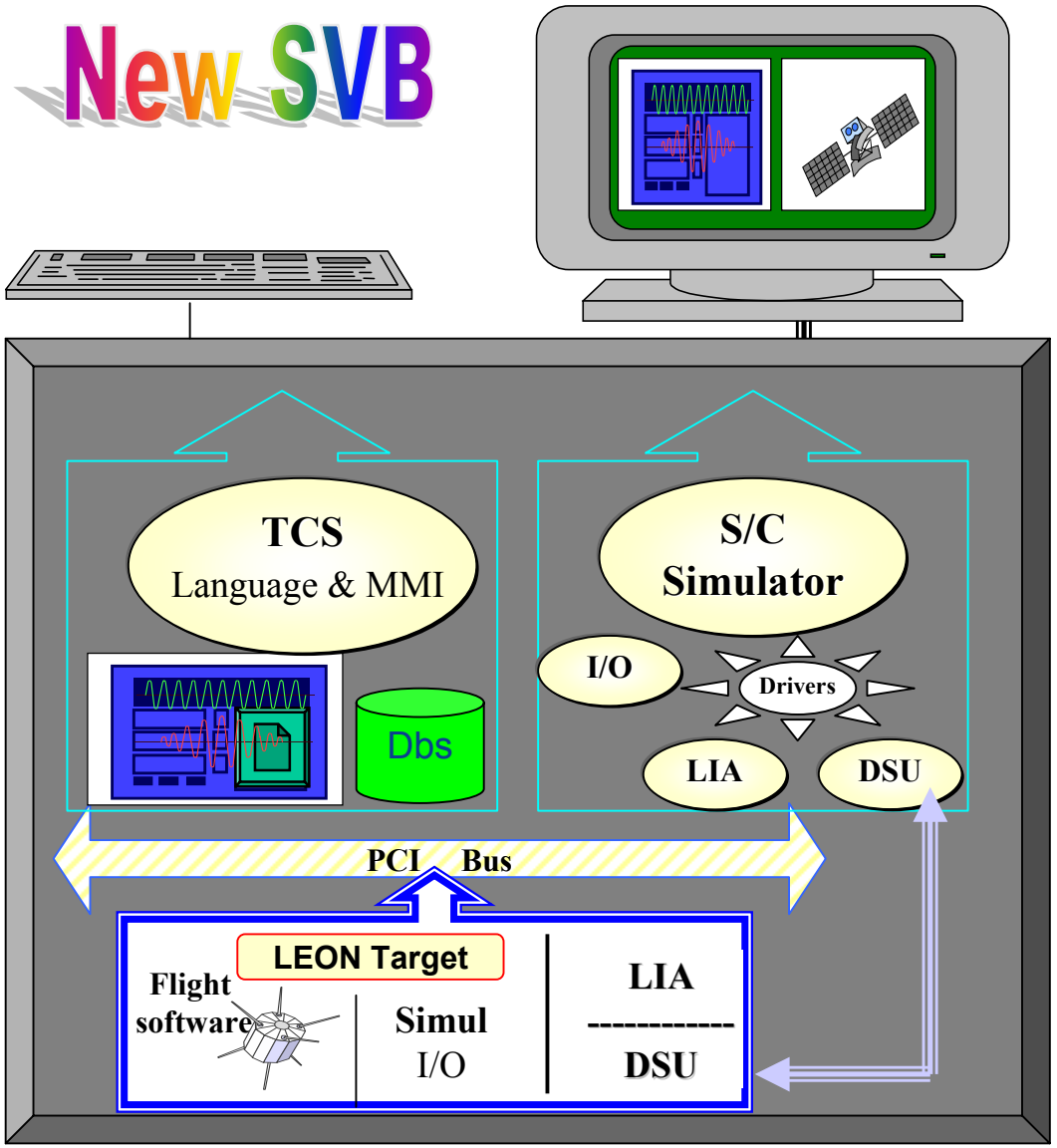


## Target

- FPGA board in PCI bus
- VHDL+I/O modelled
- Debug through DSU line
- LIA for non intrusive image

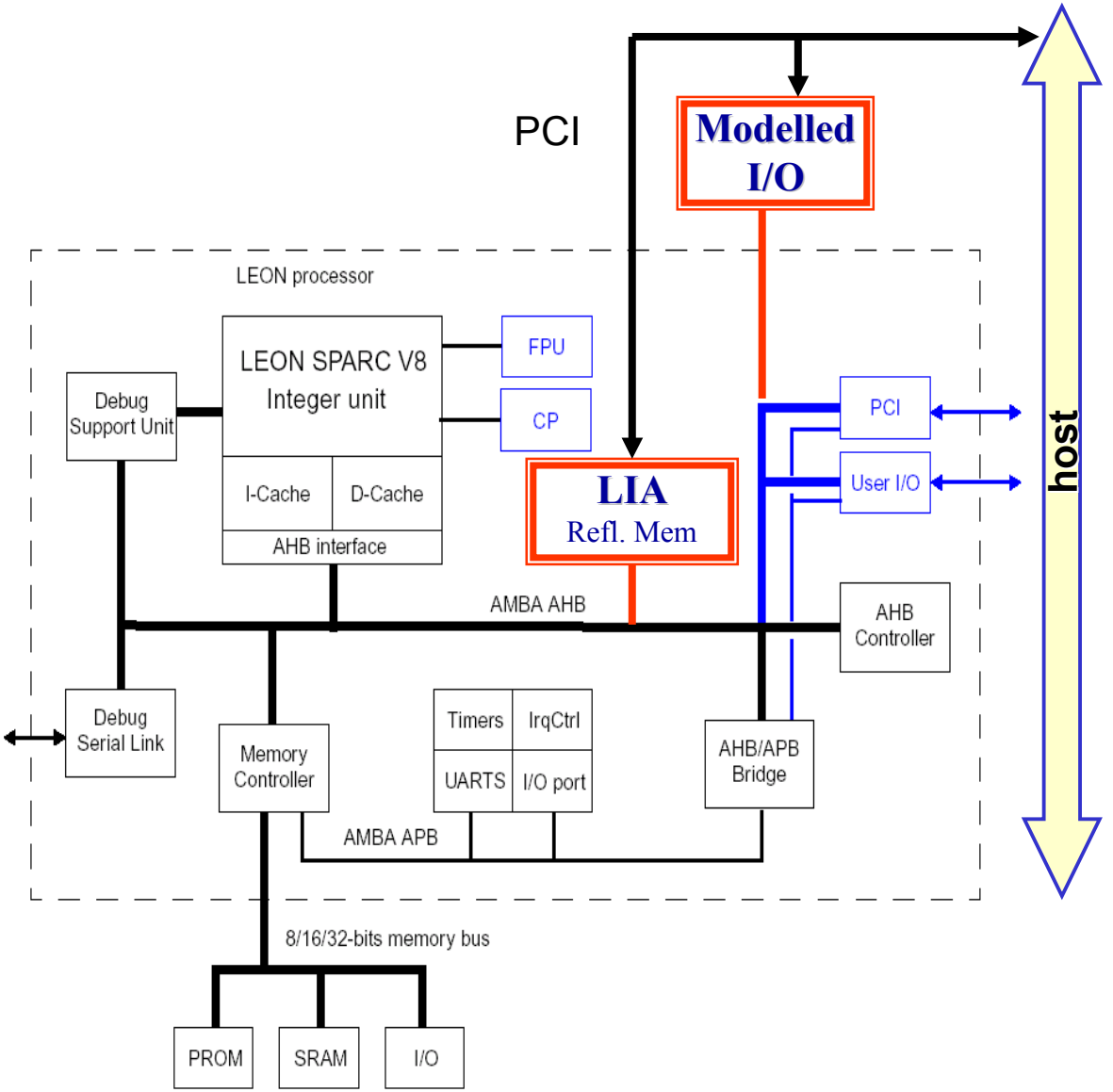
## Bench

- PC Linux Bi Processor
- Kernel 2.4.18 patched
- CPU1: for user tasks.
- CPU2: for real-time



# Zoom on LEON Target

- **Design**
  - -VHDL LEON2-1.0.8
  - -Add I/O model + feedback
  - -DSU trace extension
  - -LIA for variables non intrusive watch points & Reflective Memory
- **Performances**
  - -Depend on ground technologies
  - -Up to 65MHz with actual technologies (Vertex2-6)



# LEON SVB advantages

Intro

Past

Present

Future

- **Architecture**
  - LIA : 1000 watch points inside LEON, in an **unintrusive way.**
  - No **CPU freeze** for I/O query answers.
  - FPGA simulates target at **components level** (realistic timing).
- **Process**
  - Local retrofit for target hardware design : **ultra short cycle**.
  - **Reused as loader / debugger** for real target in latter benches.
  - Can do Close Loop on real target in single station (S/C Simulator).

# Road Map

Intro

Past

Present

Future

LEON & RTEMS on board XSV800

Linux with real-time perfs.

LEON with I/O & LIA on PCI board

DSU & LIA driver in Linux RT

LEON SVB Integration with coverage tools

Full S/C simulator TM/TC

Link with SCOS-2000

# The END

Intro

Past

Present

Future

## *Questions*


Yves Dufrenne

Software Avionics Facilities expert
IA55

E-Mail:
yves.dufrenne@astrium-space.com